SECURITY ANALYSIS SYSTEM TO DETECT THREATS ON A SIP VOIP INFRASCTRUCTURE ELEMENTS

Filip REZAC[1], *Miroslav VOZNAK*[1], *Karel TOMALA*[1], *Jan ROZHON*[1], *Jiri VYCHODIL*[1]

[1]Department of Telecommunications, Faculty of Electrical Engineering and Computer Science, VSB – Technical University of Ostrava, 17. listopadu 15, 708 33 Ostrava, Czech Republic

filip.rezac@vsb.cz, miroslav.voznak@vsb.cz, karel.tomala@vsb.cz, jan.rozhon@vsb.cz, jiri.vychodil@vsb.cz

Abstract. *SIP PBX is definitely the alpha and omega of any IP telephony infrastructure and frequently also provides other services than those related to VoIP traffic. These exchanges are, however, very often the target of attacks by external actors. The article describes a system that was developed on VSB-TU Ostrava as a testing tool to verify if the target VoIP PBX is adequately secured and protected against any real threats. The system tests the SIP element for several usually occurring attacks and it compiles evaluation of its overall security on the basis of successfull or unsuccessfull penetrations. The article describes the applications and algorithms that are used by system and the conclusion consists recommendations and guidelines to ensure effective protection against VoIP PBX threats. The system is designed as an open-source web application, thus allowing independent access and is fully extensible to other test modules.*

Keywords

SIP server, Safety Test System, Flood Attack, scanning, monitoring, SPIT, countermeasures, data manipulation.

1. Introduction

Systems designed to test and monitor networks or other components are quite wide-spread these days. Examples of the principle ones are Nessus [1], Retina [2], Snort [3] and other. The majority of these systems allows for testing the whole network infrastructures and protocols used for communication between components. None of these solutions, however, enables a complex testing of VoIP infrastructure and SIP servers which are the key and most vulnerable component of the network. The system we developed, under a working title SPT (SIP Penetration Testing), was designed as a penetration tests simulator for SIP servers. Based on the analysis of intersections, the person who initiated the testing ("the tester") receives feedback in the form of test results, as well as recommendations on how to mitigate potential security risks that were discovered. The advantage of this solution is that the system simulates real attacks from the external network, i.e. the system does not need to be placed in the same network as the target component DUT (Device under Test). This is frequently one of prerequisites to be able to use other testing tools. The SPT system was implemented as a web application accessible through a standard web browser and therefore independent on the operation system's platform. Authentication will be done using the SSO (Single Sign-On) service - Shibboleth [4]. This should also prevent the system being used for other than testing purposes. Once signed in, the tester enters the required data into a web form and chooses tests to be run. The output of the application once the tests have been completed is an e-mail report to the tester.

This paper contains the results of the tests; and in case some penetrations were successful it also contains recommendations and measures to mitigate such attacks in the future. Figure 1 illustrates the concept of the SPT system. The following chapter describes individual testing methods in detail, their implementation, algorithms used and the impact on the target SIP server.

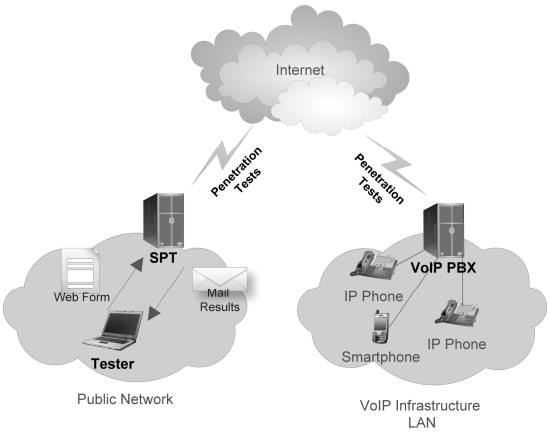

Fig. 1: SIP penetration tests system scheme.

2. Platforms, Algorithms and their Time Evaluation

Although the system is primarily designed for penetration tests on SIP servers, in reality it can perform full-scale attacks on a particular component and provide feedback on it to the tester. Thus, it is necessary to ensure that the developed system cannot be abused by a third party. The system was designed as a LAMP (Linux, Apache, MySQL, PHP) server [8] and its complete administration including the installation is carried out via a web interface. For reasons stated above, the system will only be accessible to authorised persons once they pass through the authentication. First the tester fills in the IP address or domain name of the central SIP server and the email address to which the test results will be sent. Using checkboxes, the tester may define the range of the modules offered for testing. Individual modules are described below in detail.

2.1. Scanning and Monitoring Module

In order to be able to carry out an efficient and precise attack on a SIP server, the potential attacker needs to find as much information as possible about a particular component. This is why we first developed a Scanning and Monitoring ("S&M") module for the SPT system, which is used to test the security of the server against attacks aimed at obtaining information by means of common and available tools (Fig. 2).

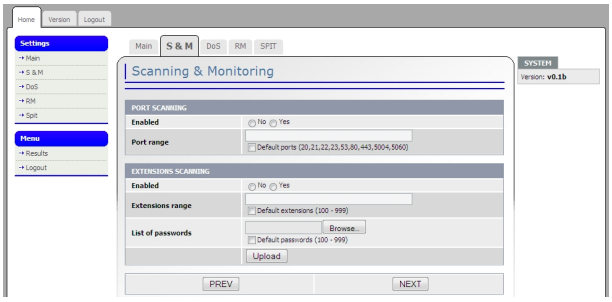

Fig. 2: SPT system – S&M module.

These tools include for instance Nmap [9] or ever more popular SIPvicious [10]. SPT system also uses these testing tools. By means of these tools, it is possible to obtain a list of listening ports or a list of user accounts created from the unsecure server. Where the server is not secured sufficiently, they can obtain even the most important - passwords to individual accounts. If the tester ticks the test to be carried out, the Nmap application is used first to establish open ports. Given the time requirements of the T_n [s] test, the testing is by default restricted only to several most frequently used ports. Using the web form, the tester can set the range of the tested ports. However the total time set for testing using Nmap is 1800 s (30 minutes). The list of available ports is subsequently included in the assessment report together with recommendations on how to minimize such ports' scanning. Another test which the SPT system can carry out aims at establishing whether SIP server's security allows for obtaining a list of user accounts. For this purpose, SIPvicious is used. By sending out OPTION and ACK requests, the application detects what accounts are defined on the SIP server. By default, the system tries the 100-999 range of accounts. Again, the tester may define own range of tested numbers E_{nr} or import a text file containing strings of alpha- numeric characters or words E_{dr}. Time required to check and create a list of T_e [s] accounts can be expressed by Eq. (1) where c = 0,02603 is a time constant obtained by repetitive measurements on a sample of 1000 potential accounts on different target SIP servers [7].

$$T_e = (E_{nr} + E_{dr}) \cdot c \,. \tag{1}$$

Number of valid accounts E_{valid} is derived from Eq. (2) where $E_{invalid}$ is the number of accounts that have been reviewed by the system but not defined on the SIP server.

$$E_{valid} = (E_{nr} + E_{dr}) - E_{invalid} \,. \tag{2}$$

Once the system has tested security of the SIP server against detecting accounts, possibility to detect passwords for individual accounts is tested. Again, this testing is carried out by *SIPvicious*. Using a pre-defined range of possible numeric passwords P_{nr} or an imported text file with alpha-numeric characters or words P_{dr}, it obtains a list of passwords for individual accounts. Time requirements on this test are expressed by the following Eq. (3).

$$T_p = \left[E_{valid} \cdot (P_{nr} + P_{dr})\right] \cdot c \,, \tag{3}$$

$$T_{sm} = T_e + T_p + T_n \,. \tag{4}$$

Now we can determine the estimated time required to carry out the complete S&M test T_{sm} (4). Using the module, we can verify whether the target SIP server is sufficiently secured against such scanning and monitoring attacks.

2.2. Denial of Service Module

One of the most frequently occurring attacks is DoS (Denial of Service). In reality, it consists of several attacks with the same characteristic feature – to lock up or restrict the availability of the attacked service so that it does not function properly. Several types of DoSs [11] can be used to achieve this; our system tests the SIP server using the most frequently used one, Flood DoS. The principle of the attack is to send a large volume of adjusted or otherwise deformed packets to the target component so that it is unable to provide its core services. As a result of the attack, CPU load increases and most of the available bandwidth is consumed, resulting in the SIP

server being unable to service regular calls, or only a minimum amount of them. To generate Flood DoS, the SPT system uses two applications: udpflood [12] and inviteflood [12]. When using udpflood, the system generates UDP packets of 1400 bytes which are directed at SIP default port 5060 of the target SIP server. The tester defines the number of generated packets and the system tests whether the packets arrived at the SIP server and whether they cause some restriction of the service availability (Fig. 3). Since we know the packet's size and therefore also the size of the Ethernet framework Fs_{udp}, we can, based on the number of generated packets P_n and the bandwidth provided B_w, determine time T_{udp} [s] required to carry out the test (5).

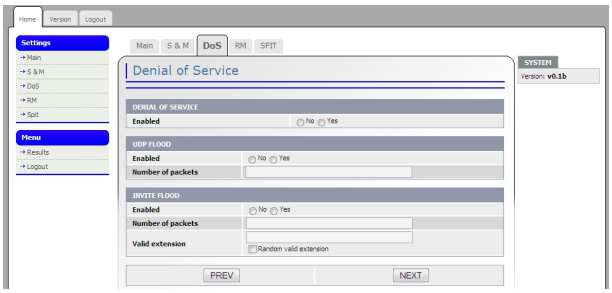

Fig. 3: SPT system - DoS module.

$$T_{udp} = \left(Fs_{udp} \cdot P_n\right) / B_w . \quad (5)$$

Table 1 provides an overview of time required for different numbers of generated packets P_n and different bandwidth B_w. When the other application, *inviteflood*, is used for testing, the system generates INVITE requests at the SIP server which are directed at an existing account. This method is very successful as most of today's SIP servers require an authentication for INVITE requests. As the INVITE requests generated by our system do not contain any authentication string, the SIP server returns SIP answer 407 Proxy Authentication Required. With the large volume of incoming requests, the load of SIP server's CPU increases. The tester can set the value of a valid account in the system manually, or it can be randomly selected from the previously obtained list of valid accounts E_{valid}. As in the previous case, we can, based on the number of generated packets P_n and the bandwidth provided B_w, determine time T_{invite} [s] required to carry out the test (6).

$$T_{invite} = \left(Fs_{invite} \cdot P_n\right) / B_w . \quad (6)$$

Figure 4 illustrates the impact of the change in bandwidth on CPU load when simulating an *udpflood* attack. The chart also clearly shows resistance of the two popular open-source SIP servers, Asterisk PBX [5] and OpenSIPS [6], to UDP Flood DoS attacks. Both centrals have been installed on the same HW of Dell PowerEdge R510 server to eliminate any potential difference in computational performance. To change bandwidths, we used HW emulator of the Simena networks. CPU load on individual centrals was measured by means of *dstat* [13].

The chart shows that OpenSIPS is many times more resistant to UDP DoS attacks than Asterisk. Total time required to carry out DoS tests T_{dos} is determined as follows (7).

Tab.1: Udpflood attack time duration with different bandwidth and number of generated packets.

Number of packets - Pn	Bandwidth [Mbit·s^{-1}] and the attack time Tudp [s]			
	10	25	50	100
100 000	113,12	45,25	22,63	11,31
200 000	226,24	90,50	45,26	22,62
300 000	339,36	135,75	67,89	33,93
400 000	452,48	181	90,52	45,24
500 000	565,60	226,25	113,15	56,55
600 000	678,72	271,5	135,78	67,86
700 000	791,84	316,75	158,41	79,17
800 000	904,96	362	181,04	90,48
900 000	1018,08	407,25	203,67	101,79
1 000 000	1131,20	452,5	226,3	113,1

$$T_{dos} = T_{udp} + T_{invite} . \quad (7)$$

Results and success rate of DoS tests carried out are included in the report for the tester.

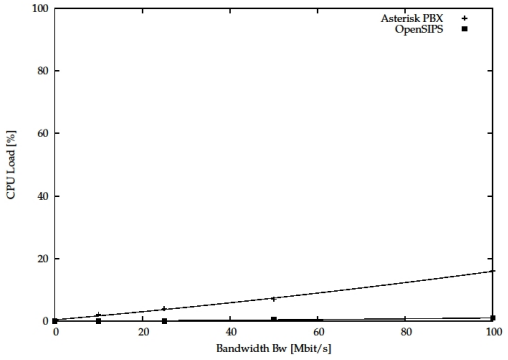

Fig. 4: Impact of change in bandwidth on CPU load in case of *udpflood* attack.

2.3. Registration Manipulation Module

Once the potential perpetrator obtains information about existing accounts, he can manipulate these accounts quite easily. The SPT system we developed can also test SIP servers' security, i.e. measures against manipulating the registration, see Fig. 5.

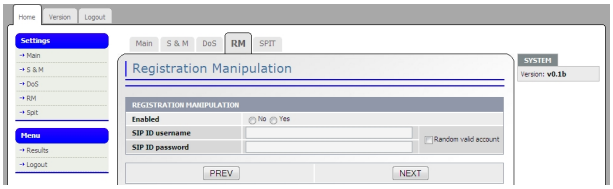

Fig. 5: SPT system - RM module.

To carry out this test, the system uses *reghijacker* [12] which substitutes the legitimate account registration with a fake, non-existing one. This type of attack can easily be expanded to a so called MITM, Man-in-the-Middle [11]. In this attack, a non-existent user is substituted by a valid SIP registration and all incoming signaling and media to the legitimate registration will be re-directed to the newly created registration. In this case, the tester needs to define the value of the SIP account which is to be stolen in the system and where authentication of REGISTER request is allowed, also a password to this account. Where the tester fails to define these values, the system automatically assigns an account and its password from the list created while scanning and monitoring the central. Time required to carry out the test T_{rm} is insignificant compared to operational times of other modules.

2.4. SPIT Module

Today, one of the most popular attacks on the Internet is spam. It is estimated that spams account for 80 – 90 % of total attacks on the Internet. Security experts predict that Spam over Internet Telephony (SPIT) will be a major threat in the future. The level of annoyance is even greater than with classical spam. Our team had developed SPITFILE [14] which served as a testing tool while developing security against such type of attacks. The SPT system uses the core of this application, together with *Sipp* [14], to simulate a SPIT attack on the target SIP server (Fig. 6). In the form, the tester defines the value of a valid SIP account – the called party to which the SPIT call will be directed and then the value and password to a valid SIP account – the caller through which the call will be initiated. Where the tester fails to define these values, the system automatically assigns an account and an appropriate password from the list created while scanning and monitoring the central.

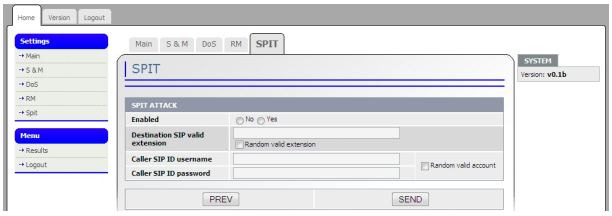

Fig. 6: SPT system - SPIT module.

If the attack was successful, a SIP call is initiated from the caller's account, and the end device with the registered account of the called party starts ringing. Once the call is answered, a pre-recorded message is played and the call terminated. Time required to carry out the test T_{spit} is determined by the length of the pre-recorded message. The final report on penetration tests which the tester receives via e-mail, will, besides information on all previous tests, also contain an analysis and success rate of the SPIT module's test.

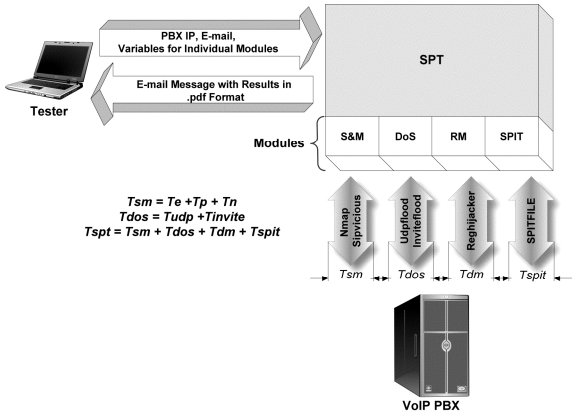

Fig. 7: Division of the SPT system into individual modules.

Figure 7 illustrates the division of the SPT system into individual modules and shows time intervals necessary to carry out individual tests in respective modules. Time requirements of the whole SPT system can be expressed by Eq. (8). Its value depends on many factors and can radically change in accordance with the type of tests requested by the tester. Its value is for reference only.

$$T_{spt} = T_{sm} + T_{dos} + T_{rm} + T_{spit} . \quad (8)$$

3. Platforms, Algorithms and their Time Evaluation

Although the SPT system is still in the phase of intensive testing and development, basic operational tests of all available modules were carried out. Each test is accompanied by a short description of countermeasure's principles and methods [12] which should limit or completely mitigate potential security gaps that were revealed during SIP servers' testing.

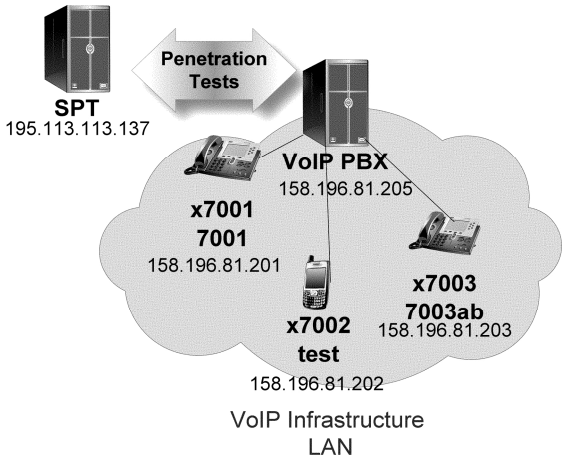

Fig. 8: SIP penetration tests system testbed.

Figure 8 describes the basic testing topology. The system is denoted as SPT and Asterisk as VoIP PBX. Asterisk was installed at Dell PowerEdge R510 server.

3.1. Scanning and Monitoring Module Testing and Countermeasures

The first step was to enter IP address of SIP server and e-mail address to which the report will be sent. Next, the S&M module and subsequently *Nmap* and *SIPvicious* applications were launched. Values for *Nmap* were set by default, value of E_{nr} for *SIPvicious* was set to range between 1000-9999. The device found all three registered accounts E_{valid} 7001-7003 and listed open TCP and UDP ports at Asterisk. Once P_{nr} was set to 7001-7003 and a text file P_{dr} containing test and 7003ab string, the test to obtain passwords to individual accounts was also successful (Fig. 10a). Total time incurred on testing module is $T_{sm} \cong 235$ s. Part of the report that is delivered to a tester's email is shown on Fig. 9. If we had to protect and prevent SIP server from scanning and monitoring, then an implementation of firewall is the effective solution or an intrusion detection system that is able to distinguish scanning and monitoring. The next effective solution is to divide the network logical infrastructure into VLANs and decompose the provided services into more physical servers (TFTP, HTTP servers). The prevention of accounts and passwords detection is difficult, moreover, the tools for detection apply the standard SIP methods and is not trivial to distinguish legitimate behaviour from an attack. In this case, it is recommended to divide the infrastructure into individual VLANs so that the detection for intruder is as difficult as possible.

```
Result of Scanning and Monitoring test
================================================================
Tested Device: 158.196.[illegible]:5060
Agent On Device: Asterisk PBX
Port Scan Results:
[20/tcp] => filtered
[21/tcp] => filtered
[22/tcp] => filtered
[23/tcp] => filtered
[53/tcp] => filtered
[80/tcp] => filtered
[443/tcp] => open
[5004/tcp] => filtered
[5060/tcp] => closed
[20/udp] => open|filtered
[21/udp] => open|filtered
[22/udp] => open|filtered
[23/udp] => open|filtered
[53/udp] => open|filtered
[80/udp] => open|filtered
[443/udp] => open|filtered
[5004/udp] => open|filtered
[5060/udp] => open|filtered

Port 443/tcpis open!!

System found extensions:
[[illegible]01] => reqauth
[ 02] => reqauth
[ 03] => reqauth

System does not found any valid password for extensions.
```

Fig. 9: Example of the SaM module test results that are sent to Tester's email.

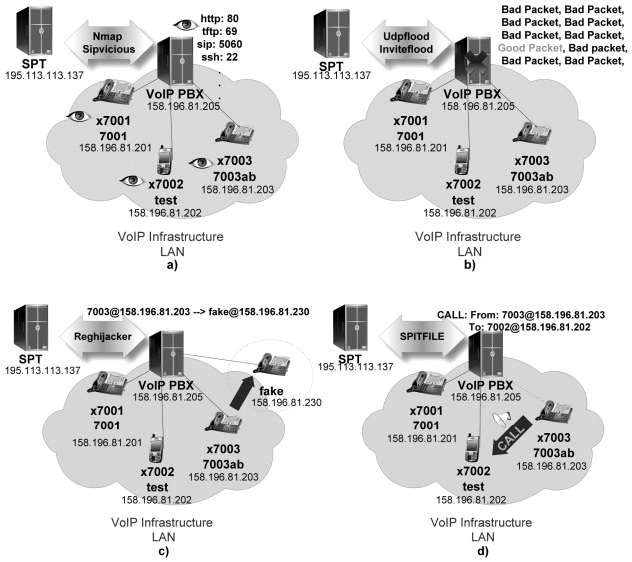

Fig. 10: SIP penetration tests System testbed. a) scanning and monitoring tests, b) DoS tests, c) registration manipulation tests, d) SPIT tests.

3.2. DoS Module Testing and Countermeasures

Using *udpflood*, the tester sent 500000 UDP packets directly to port 5060. Bandwidth was set to 100 Mbit/s, Asterisk processed 90 % calls. Once the test was completed, Asterisk recovered to a full operation mode. To be able to compare, we substituted Asterisk by OpenSIPS in this test. Call processing under the same attack was entirely error-free. When testing using *inviteflood* on the valid account 7001, we found out that this attack is much more destructive in terms of computational power. As early as at 100000 INVITE request when $T_{invite} \cong 9$ s, CPU load for both Asterisk and OpenSIPS reached 100 % and failed to process a single incoming or outgoing call. Once the test was completed, both centrals recovered to a full operation mode (Fig. 10b).

The way how to recognize whether a DoS attack is successful or not is given below. Before the test starts, SPT uses the ICMP protocol to measure the average response time from the exchange - T_{avg}. During testing (sending flood packets) the average response - T_{dosavg} is again tested in parallel using ICMP. In case that the T_{dosavg} is 150 times greater than T_{avg}, the test is marked as successful. A necessary condition for DoS test is to support an ICMP protocol. The principle of detection is shown in Fig. 11.

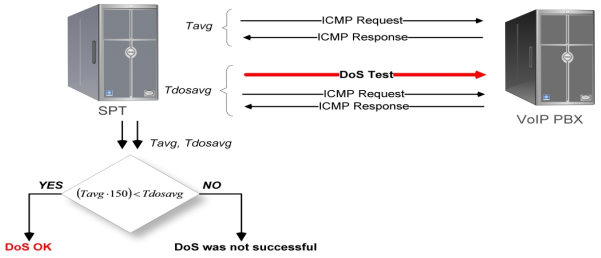

Fig. 11: Principle of DoS test detection.

The possibilities on how to protect from Flood DoS attacks, are the following: to divide the network infrastructure into separate VLANs, to have in use solely TLS, to implement L2 network elements with DoS detection or to apply SIP firewall that can detect DoS attacks and minimize their impact.

3.3. Registration Manipulation Module Testing and Countermeasures

When testing possibility for registration manipulation, we entered values of account 7003 and its password 7003ab manually into the SPT system. Once the test was completed, we established whether the attack was successful using the following process. SPT sends a SIP INVITE message to the test machine. Once a message is sent, the system waits for the type of response that comes. Based on this response SPT assess whether the test was (attack), successful or not. When system receives **180 Ringing** response the tested attack has not been successful, if a **503 Service Unavailable** response arrives, the test was implemented successfully (see Fig. 12).

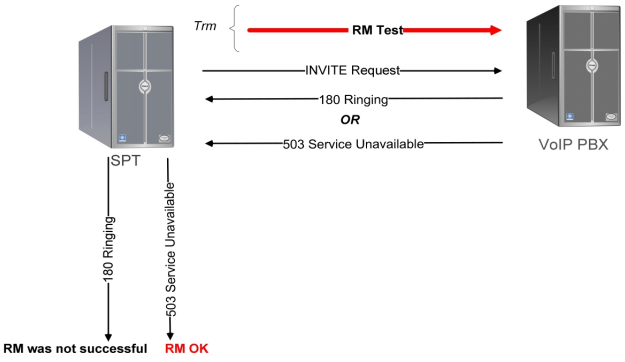

Fig. 12: Principle of RM test detection.

The aim of the attack was to de-register account 7003 and to direct all incoming calls to a fake account which does not exist. Thus, calls were terminated as unconnected. The call to 7003 was not put through, see Fig. 10c. The TCP protocol is recommended at transport level to prevent a registration hijacking because the manipulation with TCP requires higher complexity. Next option, how to minimize this threat, is to use REGISTER message authentication. We could decrease the registration interval, as well, it is quite simple but effective.

3.4. SPIT Module Testing and Countermeasures

As stated above, we used SPITFILE application, developed by authors of the paper, to test the PBX vulnerability against SPIT attacks. The tester manually inserts value of a valid account 7002 on which a SPIT attack was to be initiated, as well as the value of a valid account 7003 and password to it (7003ab) which was supposed to initiate the SPIT call. Once the test was launched, SPITFILE registered on the participant 7003 and then started to generate a call to account 7002. The end device registered on 7002 began ringing, and once the call was answered, a recording with an advertisement was played (Fig. 10d). The system detects the success of SPIT attacks as follows: during test, SPT monitors the responses of SIP VoIP PBX and based on the them, system defines whether the SPIT test was successful or not. System performs 5 consecutive SPIT calls, and all INVITE messages must be answered with **180 Ringing** response. If SPT receives any other response, SPIT test failed. Detection process can be seen in Fig. 13.

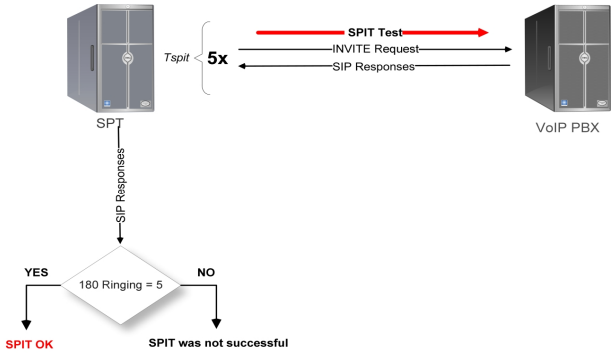

Fig. 13: Detection of SPIT test.

A few methods exist on how to restrict the SPIT propagation, which are more or less efficient, but their combination bring quite strong protection against the type of attack. Such methods include various forms of lists or voice testers based on CAPTCHA which are particularly effective against caller bots. Authors developed own solution ANTISPIT [14] that exploits the specific human behavior and automatically modifies the Blacklist table without participation of called party, the approach is based on the statistical Blacklist.

4. Conclusion and Future Work

The aim of the authors was to develop a tool to carry out penetration tests on SIP servers. The system that was designed and implemented consists of several modules that are able to generate selected types of attacks which the authors deem most popular. The system then analyses to what extent is the target component secured, drafts assessments containing tests' results and proposes factual recommendations to ensure security against the threat concerned. The assessment report is sent as a text document to an e-mail. The system is currently under intensive testing. It is planned that in the future, it will be extended to include other testing modules and functions such as for instance testing of the whole VoIP infrastructure and heavy testing of individual components.

Acknowledgment

This work has been supported by the Ministry of Education of the Czech Republic within the project LM2010005.

References

[1] ROGERS, R. *Nessus Network Auditing. Syngress*. 2. Burlington: Elsevier, 2008. pp. 433. ISBN 978-1-59749-208-9.

[2] CHOCHELINSKI, R.; BARONAK, I. Private Telecommunication Network Based on NGN. In *32nd International Conference on Telecommunications and Signal Processing*, 2009, Dunakiliti, HUNGARY, pp. 162-167. ISBN 978-963-06-7716-5.

[3] BATES, J.; GALLON, C.; BOCCI, M.; WALKER, S; TAYLOR, T. *Converged Multimedia Networks*. Wiley, 2006. pp. 364. ISBN 978-0-470-02553-6.

[4] VOZNAK, M. *Voice over IP*. VSB-Technical University of Ostrava: College Textbook, 1st. ed., Ostrava, 2008.

[5] WINTERMEZER, S.; BOSCH, S. *Practical Asterisk 1.4 and 1.6: From Beginner to Expert*. Addison-Wesley Professional; 1 edition, 2009. ISBN 978-0-321-52566-6.

[6] GONCALVEZ, F. *Building Telephony Systems with OpenSIPS 1.6*. Packt Publishing, 2010. pp. 274, ISBN 978-1849510745.

[7] SISALEM, D.; FLOROIU, J.; KUTHAN, J.; ABEND, U.; SCHULZRINNE, H. *SIP Security*. Wiley, 2009. pp. 350. ISBN 978-0-470-51636-2.

[8] LEE, J.; WARE, B. *Open Source Development with LAMP: Using Linux, Apache, MySQL, Perl, and PHP*. Addison-Wesley Professional, 2002. pp. 496. ISBN 0-785-342-77061-2.

[9] LYON, G. F. *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Nmap Project, 2009.

[10] VOZNAK, M.; ROZHON, J. SIP Infrastructure Performance Testing. In *9th International Conference on Telecommunications an Informatics*, Catania, Italy, 2010, pp. 153-158. ISBN 978-954-92600-2-1.

[11] REZAC, F.; VOZNAK, M.; RUZICKA, J. Security Risks in IP Telephony. In *CESNET Conference 2008*, 2008, Prague, pp. 31-38. ISBN 978-80-904173-0-4.

[12] ENDLER, C.; COLLIER, M. *Hacking Exposed VoIP: VoIP Security Secrets and Solutions*. McGraw-Hill Companies, 2007. pp. 539. ISBN 978-0-07-226364-0.

[13] PRAVDA, I.; VODRAZKA, J. Voice quality planning for NGN including mobile networks. In *12th International Conference on Personal Wireless Communications (PWC 2007)*, 2007, Prague. pp. 680. ISBN 978-0-387-74158-1.

[14] VOZNAK, M.; REZAC, F. The implementation of SPAM over Internet telephony and defence against this attack. In *TSP 2009: 32nd International Conference on Telecommunications and Signal Processing*, Dunakiliti, HUNGARY, Aug 26-27, 2009, pp. 200-203. ISBN 978-963-06-7716-5.

About Authors

Filip REZAC was born in 1985. In 2007, received a Bachelor title in VSB-Technical University of Ostrava, Faculty of Electronics and Computer Science, Department of Informatics. Two years later he received the M.Sc. title focused on mobile technology in the same workplace. Currently in the doctoral study he focuses on Voice over IP technology, Network Security and Call Quality in VoIP.

Miroslav VOZNAK was born in 1971. He holds position as an associate professor with Department of Telecommunications, VSB-Technical University of Ostrava, Czech Republic. He received his M.S. and Ph.D. degrees in telecommunications, dissertation thesis "Voice traffic optimization with regard to speech quality in network with VoIP technology" from the Technical University of Ostrava, in 1995 and 2002, respectively. Topics of his research interests are the next generation network, IP telephony, speech quality and network security.

Karel TOMALA received a Bachelor title in VSB-Technical University of Ostrava, Faculty of Electronics and Computer Science, Department of Electronics and Communications in 2007. Two years later he received the M.Sc. title on the Department of Telecommunications. Currently in the doctoral study he focuses on Voice over IP technology, High Availability on Telecommunication Networks and Embedded Solutions.

Jan ROZHON received his M.Sc. degree in telecommunications from VSB – Technical University of Ostrava, Czech Republic, in 2010 and he continues in studying Ph.D. degree at the same university. His research is focused on performance testing of NGN. In 2010, he received rector's appreciation for his diplomathesis.

Jiri VYCHODIL was born in 1984 in Olomouc. In 2007, he received bachelor degree at VSB-Technical University of Ostrava, Faculty of Electronics and Computer Science, Department of Telecommunications. In 2009 he received M.Sc. degree at the same department. Now, he is a Ph.D. student. His research is focused on IP telephony and computer networks.